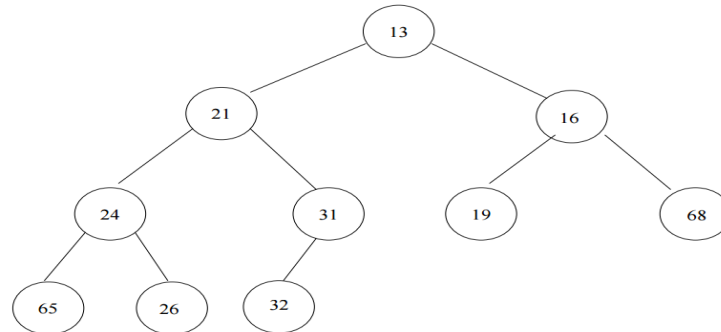


Ejercicios Estudio Solemne 3 - Estructuras de Datos UNAB 2017-1

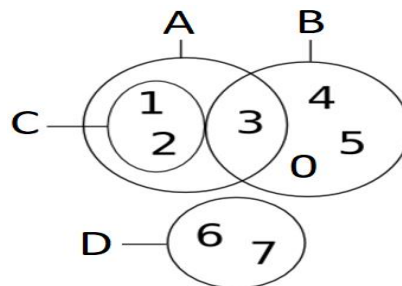
- 1) Muestre el resultado de insertar 2,1,4,5,9,3,6,7 en un árbol AVL inicialmente vacío. Dibuje el árbol paso a paso indicando el tipo de rotación aplicada en cada caso.
- 2) Aplique **heap sort** a los siguientes números, para ordenarlos de menor a mayor:
10, 8, 26, 28, 3, 5, 30, 2, 4, 11.
- 3) Dado el montículo (Heap) siguiente, traspase a un arreglo e indique los pasos a seguir para ordenar el arreglo usando heap-sort. Ejecute los pasos que ha explicado



- 4) Dado el arreglo siguiente:

1	2	3	4	5	6	7	8	9	10	11	12
40	20	70	100	60	10	120	20	50	30	10	80

- a) Crear un Heap-Min.
 - b) Aplique el algoritmo heap sort para ordenar los números del Heap generado en a).
- 5) Dada la figura siguiente:



- a) Si usted conoce los conjuntos A y B ¿con qué operaciones se obtienen los conjuntos C y D?. Fundamente su respuesta realizando dichas operaciones paso a paso.
 - b) Indique cuáles serían los conjuntos A y B si éstos se representan en un arreglo de bits (con el número más significativo a la izquierda). Obtenga C y D utilizando sólo operaciones sobre conjuntos de bits. Realice las operaciones paso a paso.
 - c) Usando las funciones de la librería **set.h**, vista en el laboratorio, que permite ejecutar operaciones sobre conjuntos, implemente en lenguaje C el main con lo siguiente:
 - Crear los conjuntos A y B e inserte los valores correspondientes.
 - Muestre (imprimir) los conjuntos A y B en su representación binaria.
 - Realice las operaciones $A \cap B$, $A \cup B$ y muestre sus resultados en representación binaria.
 - Que indique (a través de un mensaje) si el $6 \in$ o no al conjunto A.
 - Elimine el 0 del conjunto B y muestre el resultado de B en representación binaria.
- Asuma que todas las funciones del **set.h** ya están implementadas, por lo tanto, en el main sólo debe ejecutar las funciones necesarias para obtener lo solicitado.

6) Dados los siguientes conjuntos de hormonas alteradas en varios pacientes:

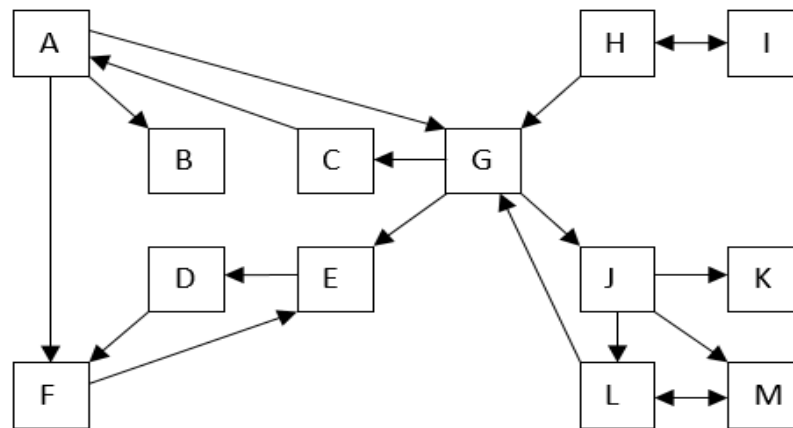
HOR1 = { 'melanina', 'oxitocina', 'dopamina' }

HOR2 = { 'testosterona', 'melanina' }

HOR3 = { 'calcitonina', 'estradiol' }

- Represente los conjuntos anteriores en conjuntos decimales. Explique muy bien su respuesta.
- Considerando una representación de conjuntos de bits, responda lo siguiente: ¿Comparten alguna hormona el conjunto HOR1 y el HOR2? ¿Y el HOR1 y el HOR3? ¿Es el conjunto HOR2 un subconjunto del HOR1? ¿Son el HOR2 y el HOR1 y el HOR2 y el HOR3 disjuntos? ¿Cuál es el conjunto de todas las hormonas?. Desarrolle paso a paso las operaciones **en bits (no debe realizar operaciones en conjuntos decimales)** necesarias para responder y concluir cada pregunta.
- Usando sólo operaciones sobre conjuntos de bits agregue el elemento 'melamina' al conjunto HOR3 y luego elimine 'estradiol' al mismo conjunto.
- ¿existe el elemento estradiol en el conjunto HOR1?. Realice paso a paso las operaciones sobre el conjunto de bits para responder la pregunta.

7) Recorrer en profundidad y anchura el siguiente grafo. Entregue la salida y el árbol de extensión en cada caso.



8) Encuentre los árboles de extensión con todas las aristas:

- por búsqueda en profundidad y
- por búsqueda en amplitud (anchura)

del siguiente grafo no dirigido buscando en los casos con alternativas para avanzar en orden numérico partiendo desde el nodo 1:

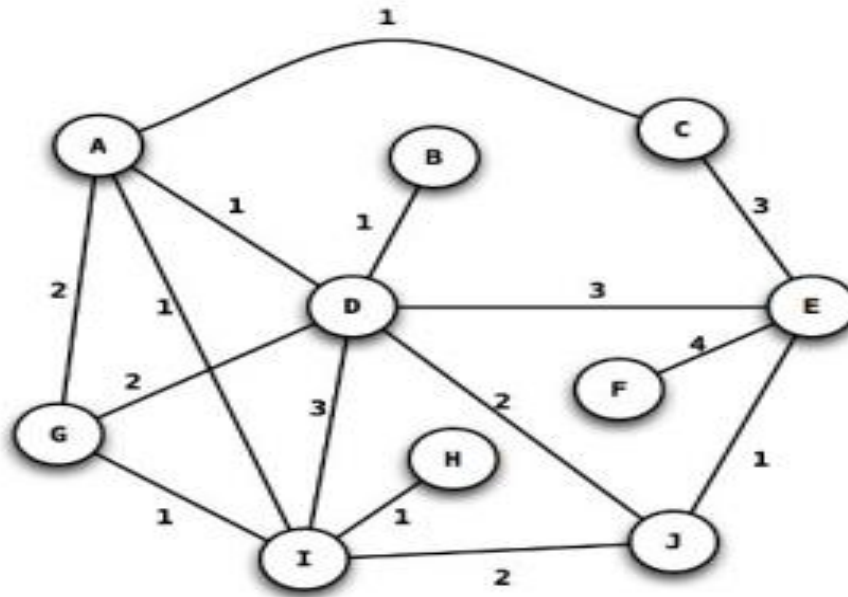
1-2, 1-3, 1-4, 2-3, 2-5, 2-8, 3-6, 3-9, 4-7, 4-10, 4-11, 5-6, 5-8, 6-9, 7-9, 10-11, 10-12.

9) Aplique los algoritmos de Prim y de Kruskal para obtener los árboles de extensión de costo mínimo del siguiente grafo no dirigido dado por nodo1-nodo2-costo:

1-2-1, 1-4-3, 1-6-4, 2-3-5, 2-5-3, 2-6-2, 3-4-3, 3-5-2, 3-6-1, 4-5-1, 5-6-4

10) Para el siguiente Grafo:

- Recorrer el grafo en profundidad y por anchura (por separado) considerando que el nodo inicial es A y el criterio principal es por costo y sub-criterio orden alfabético.
- Aplicar los algoritmos de Kruskal y Prim para encontrar un árbol de cobertura de peso mínimo del grafo G. En cada caso, enumerar las aristas en el orden en el que son seleccionadas. Para el algoritmo de Prim especifique la lista L con las aristas del grafo ordenadas de menor a mayor peso.



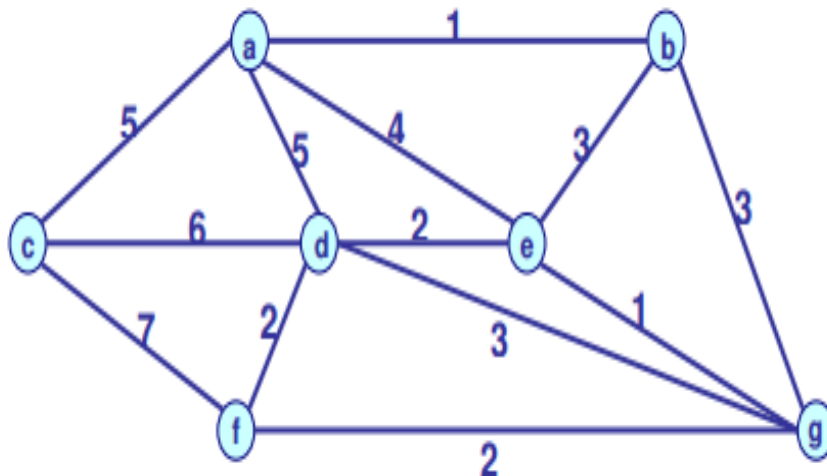
- 11) De acuerdo al siguiente grafo dirigido con 6 nodos cuyos datos para sus 9 aristas se entrega de la siguiente forma: (nodo1-nodo2-costo):

1-2-10, 1-4-50, 2-3-15, 2-4-20, 3-4-20, 3-5-40, 4-5-10, 4-6-15, 5-6-20

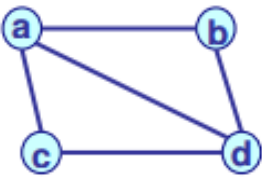
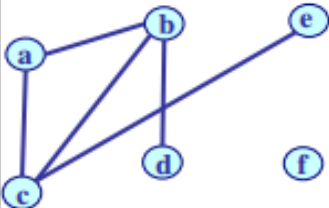

Aplique los algoritmos de Dijkstra y de Lennon Ford (por separado) para calcular las longitudes de los caminos más cortos desde el nodo 1 hacia todos los demás nodos del grafo y los caminos más cortos desde el nodo 1 hacia todos los demás nodos. ¿Los recorridos y costos que entrega cada algoritmo son los mismos?

NOTA: Represente la ejecución de los algoritmos **SÓLO** a través de los métodos entregados en clases (distancia acumulada en cada nodo en Dijkstra y tabla por iteración en Lennon Ford).

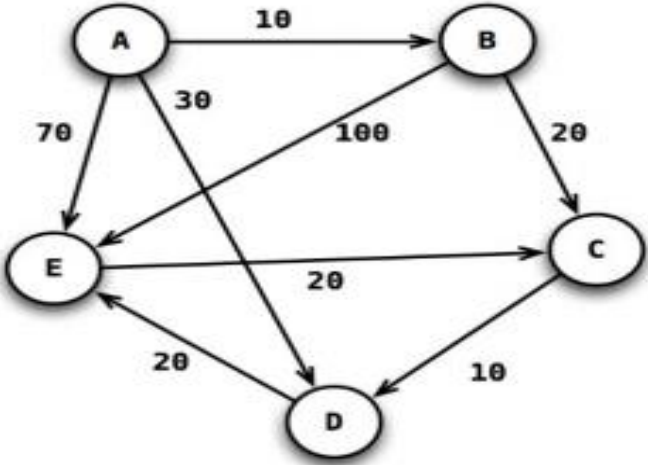
- 12) En el grafo de la siguiente figura se muestra una red de computadores que se quiere conectar, los vértices representan los computadores y las aristas las líneas de transmisión a considerar para conectar algunos pares de ellos. Cada arista tiene un peso que indica el costo de conectar esa línea específica. Se pide conectar todos los computadores con el menor costo posible.



13) Completar la siguiente tabla:

Grafo	Matriz de adyacencia	Lista de adyacencia
		
		
		

14) Dado el grafo dirigido que se muestra la siguiente figura, se pide: Calcular, mediante los algoritmos de Dijkstra y Lennon Ford (por separado), el camino mínimo desde cualquier vértice (fuente) al vértice (destino) del grafo. Mostrar para ello, la secuencia de matrices de costos (D). Represente la ejecución de los algoritmos **SÓLO** a través de los métodos entregados en clases (distancia acumulada en cada nodo en Dijkstra y tabla por iteración en Lennon Ford).



15) Dado el siguiente grafo escrito como lista de adyacencia:

a) Escriba el grafo como matriz de adyacencia y Dibújelo.

b) Realice el recorrido en Profundidad y en Extensión iniciando desde el nodo a, anote los pasos y orden que recorrió los nodos.

```
a → b → d → //
b → c → e → //
c → e → //
d → b → e → //
e → c → //
```

- 16) Dada la siguiente matriz de costos, utilice el algoritmo de Kruskal para obtener el árbol de camino mínimo:

	a	b	c	d	e	f	g	h	i	j
a	0	33	10	56	-	-	-	-	-	-
b	33	0	-	13	21	-	-	-	-	-
c	10	-	0	23	-	24	65	-	-	-
d	56	13	23	0	51	-	20	-	-	-
e	-	21	-	51	0	-	17	35	-	-
f	-	-	24	-	-	0	40	-	72	-
g	-	-	65	20	17	40	0	99	45	42
h	-	-	-	-	35	-	99	0	-	38
i	-	-	-	-	-	72	45	-	0	83
j	-	-	-	-	-	42	42	38	83	0

- 17) Un grafo no dirigido con un máximo de N nodos con las etiquetas 0 a N-1 está dado por listas de adyacencia con la siguiente estructura:

```

struct nodo    {
    int valor;
    struct nodo *sig;
};
struct nodo *arregloListas[N];

```

Programe en C la función void agregarArista(int a, int b), que agrega la arista entre nodo a y el nodo b. Se supone que estos dos nodos forman parte del grafo.

- 18) Implementar una función que reciba como parámetro una matriz (de un grafo dirigido) y un entero n (que representa un nodo del grafo) y que retorne la cantidad de vecinos a los que se puede acceder desde n.